ABSTRACT

As the impetus to extend computer and communications as far possible into the battlefield continues to grow, configuration management of these computer and communication networks becomes a challenging task. The challenges stem mainly from the volatile connectivity (due to network element failures) and sporadic mobility (due to the dynamics of the battlefield theatre). This paper proposes a self-configuration management system for such battlefield networks. The issues addressed and resolved by the proposed design are: (a) self-configuration - the ability to adapt and configure the network without human intervention when the network elements relocate, and (b) robustness - the ability to provide continued service in the midst of random network facility failures.

1. INTRODUCTION

Configuration management of battlefield networks is a challenging task. This is because battlefield networks are generally characterized by volatile connectivity due to both hostile attacks on the network, and, due to mobility. Thus the communication resources in battlefield networks (such as links and/or switching nodes) can vary erratically. To provide uninterrupted communications in such a scenario, we observe that the battlefield network must have the capability of configuring and healing itself as the communication elements relocate, or even cease to exist. Further, since the failure and/or movement can occur sporadically, the network must be capable of configuring itself without human intervention, or, with the least amount of human intervention that is possible.

This paper details the design of a configuration management system that can be used in a battlefield environment of ATM based tactical networks. The issues involved are: (i) self-configuration - i.e., the ability to learn and adapt to the changing network conditions without human intervention, and (ii) robustness in the presence of failures (hence reduced network resources) - i.e., offer uninterrupted communication services in the midst of network element failures. The above issues have been resolved via the proposed self configuration management system.

In particular, the proposed self configuration management system performs the following: (a) determines a suitable initial network connectivity (topology) between “N” switches given the requirement that these switches have to be deployed in the battlefield to form the tactical backbone, (b) employs the proposed handshaking mechanism to discover, negotiate usable RF frequencies, establish, and maintain connectivity between a brigade level remote ATM switch housed in a Radio Access Peripheral (RAP) and an ATM switch in the tactical backbone, (c) adapts the PNNI signaling and routing protocol as defined by the ATM Forum for static networks to the volatile battlefield environment to establish and maintain connectivity in the battlefield network, and (d) provides connectivity/reachability by determining “an efficient route” both when a session is set up in the network, and when the RAPs relocate and/or some of the intermediate switches and/or links cease to exist.

The self configuration management system proposed in this paper handles two kinds of scenarios: (a) a scenario that corresponds to self-configuration of mobile RAPs, whose locations are random and discretized (i.e. “plug and play” as dictated by the random and dynamic battlefield conditions), and (b) a scenario that corresponds to self-healing in the core tactical backbone, wherein the switches with a given connectivity can sporadically fail and recover. Note that the proposed self configuration management system can be extended easily to accommodate “on-the-move” operations.

The contents of this paper are organized as follows. Section 2 discusses the proposed method for determining a suitable initial network connectivity given the requirement that “N” switches have to be deployed in the battlefield to form the tactical backbone. Section 3 summarizes the PNNI signaling and routing protocol identified for supporting self-configuration in the tactical backbone. Section 4 describes the design of a handshaking mechanism to discover, establish, and maintain connectivity between a switch inside a moving RAP and a core switch in the tactical backbone. Section 5 discusses the proof-of-concept simulation model constructed and Section 6 discusses the performance of the proposed self configuration management strategy. Section 7 concludes the paper and identifies possible future work.

2. NETWORK TOPOLOGY

We consider ATM-based networks wherein the ATM switches are interconnected by RF links (due to the wireless and mobile nature of a battlefield network). The network topology consists of core switches that interconnect to form
a tactical backbone network. Attached to the tactical backbone, are mobile RAP units (representative of the brigade/soldier level switches). The RAP units can attach themselves to an available core switch via certain negotiable frequencies. They are equipped with the ability to engage and disengage communications to and from a given core switch, to facilitate plug-and-play operation within the battlefield network.

To determine the initial network connectivity, we proceed as follows. The problem on hand is the following. We are given a certain number of ATM switches that need to be deployed in the battlefield to form the tactical backbone portion of the network. We first need to determine how these switches have to be deployed in the battlefield theater. In order to do this, we have developed a heuristic that determines the connectivity (presence of links) between the various switches in the following manner. Links are generated between the switches in order to satisfy the following constraints: (a) there is either a direct link or a two-hop path between every possible communicating switch pairs in the network with the constraint that the network is not fully interconnected (thus avoiding the $O(N^2)$ number of links of a fully connected network), and (b) there exists at least two permissible routes for every possible communicating switch pair. Feature (a) ensures quick reachability for a given communicating switch pair in the network (essential to maintaining fast dissemination of battlefield information), while feature (b) ensures the presence of alternative routes (essential for survivability in the presence of link and/or intermediate switch failures).

Thus, in summary, the output of the proposed heuristic is a set of links that interconnect the given ATM switches to form a certain network topology. The class of networks considered thus formulated are characterized by the following: (1) They have neither too dense (e.g., fully interconnected) nor too sparse topologies. The above helps avoid a plethora of RF links on one hand, while maintaining the presence of at least one path with a maximum path length $\leq 2$, on the other hand. (2) There exists alternative routes for every possible communicating pairs of switches in the core network, thereby facilitating rerouting to ensure survivability amidst network element failures.

### 3. PNNI SIGNALING AND ROUTING PROTOCOL

We propose that the signaling and routing protocol for the battlefield networks be based on the PNNI specifications. PNNI is developed by the ATM Forum for interfaces between private ATM switches or private ATM networks[1]. The PNNI protocol will lend itself well to battlefield networks because it enables: (a) discovery of neighbors and their link status, (b) periodic flooding of topology information (PNNI Topology State Elements-PTSEs), (c) synchronization of topology databases, and (d) user-defined event triggered flooding, among other functionalities.

In the remainder of this section, we discuss briefly how the above features are used in our proposed self configuration management system. However, before we do this, we note that since the original PNNI protocol proposed by the ATM Forum was done for stationary networks, we have performed extensions to accommodate mobility of the attached RAPs. This has resulted in the design of a specific handshaking protocol that can be used with PNNI to discover, establish and maintain connectivity with the RAPs, and is discussed in Section 4.

The core switches in the tactical backbone, use “Hello” packets to: (i) establish their identities with each of their neighboring switches, and (ii) maintain subsequent neighbor connectivity information. Recall from Section 2 that the initial neighbor connectivity in the tactical backbone is obtained via the proposed heuristic. Next, the network topology database at each core switch is updated via periodically triggered flooding of PTSEs, thus enabling all of the core switches to see a globally consistent view of the existing topology. One of the key differences between Hello message exchange and periodic flooding is that the former is only exchanged between neighboring switches while the latter is flooded to all of the network switches.

We use event triggered flooding to propagate unexpected network changes that occur in between the periodic flooding instants. To do this, we define the event set $E$ to contain the following elements, namely, $E = \{F, R, M\}$ where $F$ denotes failure (switch or link failure), $R$ represents recovery, and $M$ denotes network mobility. By defining $E$ to include RAP movements, link and switch failures, and their possible recovery, we anticipate that most of the vital network information will be propagated by event triggered flooding. Thus these event triggered flooding messages in conjunction with Hello and periodic flooding, provide an efficient self-configuration management system for the tactical network.

Regarding the routing mechanism, we employ dynamic source routing specified according to PNNI. A dynamic scheme is used because dynamic schemes have better resilience in handling unexpected network changes than pre-planned (static) schemes. Source routing is used because it does not have routing loops, which can be a problem with hop-by-hop routing. Finally, since our goal is to provide a robust self configuration management system that does not introduce unnecessary complexities, we use a
fairly simple yet efficient source routing algorithm based on minimum-cost route calculations (a Dijkstra algorithm) rather than use complex and sophisticated routing engines such as ones resulting from Markov decision theory.

Note that while the output from Section 2 determines the initial connectivity, subsequent connectivity information is maintained via the exchange of Hello, periodic flooding and event triggered flooding messages. Next, we have modified the PNNI protocol so that PTSE messages are only flooded between the core switches in the tactical backbone. Further, we propose the use of a streamlined version of the Hello messages to be exchanged between the RAP and the core switches to keep their connectivity alive, as described in Section 4.

4. RAP-CORE SWITCH HANDSHAKE PROTOCOL

In this section, we discuss a handshaking mechanism that can be utilized by a mobile RAP to discover, establish, and maintain connectivity with a core switch in the tactical backbone. The RAP is characterized by discrete randomized movements, representative of a plug-and-play scenario. Our goal in the design of a suitable discovery protocol has been governed by the need: (i) to easily incorporate the proposed RAP-core discover protocol with the existing PNNI framework, (ii) for simplicity with as little overhead as possible, and (iii) to be powerful enough to support sporadic mobility. The specific protocol that we propose is as follows:

- The core switches periodically broadcast on their omni-directional transmitters the following:
  - frequency for data communication with a RAP
  - their own co-ordinates

To reduce any unnecessary RF transmissions, it is required that only core switches that have an available transceiver partake in this RAP-Core switch handshake protocol. For each available frequency broadcast, a finite state machine (FSM) is set up in the core switch.

- Upon receipt of the above broadcast message from the core switch, a RAP unit that wishes to set up communications with the tactical backbone, does the following:
  - Obtain the data communication frequency and the co-ordinates of the first reachable core switch, and set up a directional channel for subsequent potential data communications with it.

- Fill in its own co-ordinates together with that of the target core switch with which it intends to communicate, and broadcast this back to the core switches. Observe that despite having aligned its directional channel with the target core switch, the RAP cannot transmit on this directional channel right away since the target core switch will not as yet have tuned its transmitter and set its target co-ordinates to correspond to this particular RAP.

- To ensure that only the target core switch responds to this particular RAP, the RAP inserts in its broadcast message the target core switch’s co-ordinates, which serves as a key to the intended (target) core switch. Further, since there exists the potential of multiple RAPs responding to a broadcast on one particular frequency, the RAPs also acknowledge the frequency they will use for the data communication by encoding it in their broadcast message. This information will be used by the target core switches to prevent multiple RAPs from attaching themselves to a specific core on the same frequency channel, as explained later in this section.

- Start a timer (T1) to set an upper bound for the time that a RAP will wait for a core switch to respond. If the RAP fails to hear a response within a time-out value set by this timer, it de-aligns the directional channel that it had set up earlier in this step, and awaits a fresh broadcast from a core switch.

- Upon receipt of the broadcast message from the RAP, each core switch first examines the key (i.e., the RAP’s target co-ordinates) to see if the received key matches its own co-ordinates. If it does not, the core switch simply destroys the received message. However, if the key matches with its own coordinates, the core switch extracts the frequency and co-ordinates of the RAP that has identified this switch as its target. It next checks its local copy of the advertised frequency (from its FSM) for a match with the frequency extracted from the message advertised by the RAP and performs the following:

  - If the frequencies match, the following actions are taken:
    - The core switch establishes a directional channel with the RAP that it has just heard from, and initiates a Hello process between itself and the newly aligned RAP unit. To reduce overheads, we propose the use of a

---

1. Observe however that more often than not, requirements (i)-(iii) may conflict each other.
streamlined Hello message wherein only the sending node's ID along with the Hello Timer that it proposes to use, is transmitted on this directional channel. The transmission of the ID serves the following purposes: (i) It allows the RAP to store this ID in its (the RAPs) database, which the RAP uses for the ensuing communication, (ii) it primes the path for subsequent data transfer, and (iii) it acts as an ACK to the RAP that has aligned its directional channel to the particular core switch, and thereby signals that subsequent data communication can take place via this directional channel.

— It checks to see if another transceiver is available. If yes, the transceiver frequency and switch co-ordinates will be broadcast at the next periodic broadcast interval.

- If on the other hand the frequencies do not match, the core switch destroys the message and takes no further action. The RAP will learn of this by timing out on its end, and try to establish communication with another available frequency or another core switch instead. Finally, the potential problem of two or more RAPs aligning with the same target core switch and using a single frequency is avoided by having the target core switch verify its coordinates and frequency from the message broadcast by the intended RAP. Further, the above is achieved without the need to transmit any explicit messages to signal this.

- Upon receipt of the target core switch's ID, the RAP unit does the following:
  - Send its own ID to the target core switch on its directional channel so that the target switch can update its topology database, as part of the Hello process.
  - Stop T1 timer.
  - Upon receipt of the RAP's ID, the target core switch propagates the newly attached RAP's ID to other switches by triggering an event triggered flooding message, so that all the other switches update their respective databases with this new reachability information. The core switch then continues to exchange the streamlined Hello packets with the RAP unit for as long as the two remain connected with each other.

This marks the end of the proposed RAP-Core Discovery process and the 4-way handshake mechanism. Observe that the subsequent exchange of Hello messages between the core switch and RAP enable the core switch to dedicate a transceiver to the particular RAP for the duration of the RAP's association with the given core switch. If the RAP ceases to exist, or, abruptly changes its location causing it to lose contact with its target core switch, the corresponding core switch will time out and free the transceiver so that some other RAP that wishes to communicate with it may use it. This ensures that the RF channel is not wasted by keeping it locked on to a RAP that may no longer exist due to the volatility of the environment.

5. PROOF-OF-CONCEPT SIMULATION MODEL

In this section, we briefly describe the proof-of-concept simulation model that has been developed to demonstrate the viability and salient features of the proposed self-configuration management system. A detailed call-by-call simulation model has been developed using OPNET\(^2\). Figure 5-1 displays the functional components of the developed model.

![Figure 5-1](image-url)

The tactical backbone network in Figure 5-1 is comprised of five ATM switches connected by point-to-point RF links and is used to interconnect two mobile RAPs\(^3\). The RAPs are characterized by discrete random movements. The tactical backbone connectivity is determined via the proposed heuristic outlined in Section 2. This is a separate program, requiring as its inputs the number of ATM switches to be deployed and a constraint on the maximum allowable links emanating from a single switch. The output of this program consists of a list of interconnecting links

---

2. OPNET is a discrete-event simulation tool from MIL3.
3. Due to limitations experienced with the tool regarding the size of the model, we use just five switches to demonstrate the features of the proposed self-configuration management system design. It is however to be noted that the proposed design is not limited to small networks alone.
between the various switches in the tactical backbone. The protocols discussed in Sections 3 and 4 have been incorporated into the simulation model.

With this simulation model, we provide proof-of-concept simulations that demonstrate the viability and features of the proposed design via a few sample scenarios that capture some of the battlefield dynamics. In particular, we look at scenarios when one or both RAP units relocate coupled with node failures and recovery. To contain the volume of this paper, the details of these scenarios and of the simulation model itself are provided in [4].

6. PERFORMANCE DISCUSSION

This section discusses the performance of the proposed self-configuration management system with particular emphasis on the efficiency of the proposed RAP-Core discovery mechanism. More specifically, we present a method to compute two important parameters: (a) the rap-core discovery delay (Delay\_rap\_cs), and (b) the control traffic overhead introduced by the RAP-Core discovery protocol. We emphasize the above parameters because they play an important role in the performance of the proposed self-configuration management system. While (a) sheds useful insights into the service denial (blocking) for new requests during the re-configuration period, (b) sheds information on the complexity of the proposed protocol.

6.1 Computation of rap-core discovery delay (Delay\_rap\_cs)

We define the rap-core discovery delay (Delay\_rap\_cs) as the elapsed time from when a RAP begins partaking in the RAP-Core discovery protocol, to the time it takes for all of the core network switches to register this new RAP. Delay\_rap\_cs in turn is composed of two components: (a) Discover Delay (Delay\_disc) and (b) Flooding Delay (Delay\_flood). In the discussion below, we define these two components and present a method to compute them.

**Discover Delay:** This is defined as the elapsed time from when a core switch advertises an available transceiver, to the time it takes for the newly aligned RAP's ID at the corresponding core switch on the newly established directional channel between the corresponding RAP and core, i.e., the elapsed time for the 4-way handshake described in Section 4.

**Flooding Delay:** This is defined as the time it takes to flood the ID of the newly aligned RAP amongst all of the core switches in the tactical portion of the battlefield network.

The computation of the above delays is as follows: The discover delay, Delay\_disc, is a function of the following, i.e.,

\[ Delay_{disc} = \mathcal{f}(T_{broad}, T_{proc_rap}, T_{proc_core}, T_{Q_rap}, T_{Q_core}, T_{prop}) \]  

where \( T_{broad} \) represents the periodic time interval between broadcasts from a core switch indicating the availability of a transceiver, \( T_{proc_rap} \) and \( T_{proc_core} \), respectively, represent the processing time for the discover message in a RAP unit and core switch, \( T_{Q_rap} \) and \( T_{Q_core} \), respectively, denote the queueing delays within the RAP unit and core switch, and \( T_{prop} \) denotes the message propagation delay.

In equation (1) \( T_{broad}, T_{proc_rap}, T_{proc_core} \) and \( T_{prop} \) assume some fixed values for a given discover message broadcast interval, processor type, and distance, while \( T_{Q_rap} \) and \( T_{Q_core} \) are random variables dependant on the system dynamics (call arrivals, link loading, etc.). More specifically, we have:

\[ Delay_{disc} = (T_{broad} T_{prop} + T_{prop}) + 2\times (T_{Q_rap} + T_{proc_rap} + T_{prop}) + 2\times (T_{Q_core} + T_{proc_core} + T_{prop}) \]  

In equation (2) the first term within the parenthesis represents the scenario when a RAP unit switches on just after it has missed hearing a discover message broadcasted by a core switch. (This therefore represents the worst case scenario with respect to term 1 in eqn. (2)). Thus it has to wait for the next broadcast message which will occur after \( (T_{broad} T_{prop} + T_{prop}) \) where \( T_{prop} \) denotes the small time interval by which the RAP has just missed hearing the current broadcast message. The second and third terms in equation (2) represent the time delays incurred at the RAP unit and the core switch, respectively.

To compute the next component of Delay\_rap\_cs, namely, Delay\_flood we proceed as follows. In computing Delay\_flood, we observe that Delay\_flood will be dictated by the time it takes the furthermost node with respect to the originating node in the network to obtain the flooded information. We define the furthermost node as the last node in the network to receive information of the added RAP. Thus, Delay\_flood is a function of the following:

\[ Delay_{flood} = \mathcal{f}(Hops\_furth, T_{prop}, T_{Q_core}, T_{proc_core}) \]  

where Hops\_furth is defined as the number of hops to the furthermost node. \( T_{prop}, T_{Q_core} \) and \( T_{proc_core} \) are similar as before. Hops\_furth is dependent on the interconnection topology. More specifically, we can write the following:

\[ Delay_{flood} = \sum_{i=1}^{Hops\_furth} T_{Q-core} + T_{proc-core} + T_{prop} \]  

Finally, from equations (2) and (4), we obtain the total delay in the rap-core-discovery process as:

\[ Delay_{rap\_cs} = Delay_{disc} + Delay_{flood} \]
Since many of the parameters in equation (5) are specific to a given network topology, we provide a sample calculation for an example network. Consider, for example, a fully interconnected network. Further, assume that the network is lightly loaded. The above imply, respectively, that $Hops_{-further}$ is 1, and $T_{Q_{-core}}$ as well $T_{Q_{-rap}}$ are negligible. In such a case we have $Delay_{rap_{-cs}} = 2.00002465$ secs. In computing the above, the discover message processing time at a core and rap unit is taken to equal 2.83 μ secs (based upon a message size of 1 ATM cell and an STC-3c speed link), $T_{prop} = 3.33$ μ secs (based on a radial distance of 1 km), $\delta = 2.83$ μ secs and $T_{broadcast} = 2$ secs.

From the above we note that the main delay factor in $Delay_{rap_{-cs}}$ is the discover message broadcast interval $T_{broadcast}$ which is a parameter that can be defined by the network engineer. The proposed protocol in itself is seen to contribute negligibly to the overall delay, which is an important requirement for an efficient self-configuration management system design.

Finally, observe, that $T_{Q_{-core}}$ and $T_{Q_{-rap}}$ were negligible since we assume a lightly loaded network. However, in reality, since the network will most likely be operated in a moderate to heavily loaded scenario, the above assumption may not necessarily hold, esp., $T_{Q_{-core}}$. Thus $T_{Q_{-core}}$ and $T_{Q_{-rap}}$ have to be computed. An analytic evaluation of $T_{Q_{-core}}$ and $T_{Q_{-rap}}$ involves a time and state dependent characterization of the given network. Such a characterization can be easily shown to be infeasible due to the dynamics of the routing and relative positions of the mobile RAP units. This therefore underscores the necessity for detailed simulations that capture the dynamics of the battlefield network and estimate the queuing delays at the RAP and core units accordingly. The simulation model constructed in this work (to demonstrate the viability of the proposed design) may also be used to conduct such studies, which is in fact, a proposed future work item.

6.2 Computation of the control traffic overhead due to the discovery protocol

The 4-way RAP-Core switch discovery mechanism described in Section 4 has two message types which are as follows: (1) The first two phases exchange messages which contain information about the self and target co-ordinates, and the usable frequency, which constitutes one message type i.e., Type 1 Discover message, and (2) The final two phases exchange messages which contain the sender’s ID. This constitutes the second message type, i.e., Type 2 Discover message.

We propose the use of the physical layer ATM cells for Type 1 discover messages. Based on the information content in each message type, both Type 1 and Type 2 Discover messages are each 1 cell long. With the above, the control traffic incurred per RAP-core alignment attempt is 4 cells (assuming that there is no contention between RAPs). Yet another way of examining the control traffic overhead consists of computing the control traffic in terms of bits/sec during the establishment phase (b bps), and then computing the ratio of the above to the link bandwidth also measured in bits/sec (B bps). This can be easily done by first dividing 4 by $Delay_{disc}$ to give us “b”, and then obtaining the ratio b/B. For the particular example cited, this ratio turns out to be $1.33x10^{-08}$. The above imply that the control traffic generated each time a RAP wishes to align itself with a core switch is negligible. This is again a desirable feature (i.e., a low control traffic overhead), especially given the fact that the available resources in the battlefield network are precious and can vary sporadically. Finally, as mentioned in Section 6.1, detailed call-by-call simulations can help in obtaining more accurate numbers since the $Delay_{disc}$ obtained is an approximation. The simulation model developed in this work can also be used for this evaluation.

7. CONCLUDING REMARKS

This paper proposes and discusses the design of a self-configuration management system for battlefield networks. The challenges in the design of an efficient self-configuration management system for battlefield networks stem mainly from the volatile connectivity and sporadic/abrupt movement as dictated in a battlefield theatre. The main issues addressed in this work are: (i) self-configuration and (ii) robustness, i.e., continued connectivity in the presence of sporadic network facility failures. Sections 2-5 present the details of the proposed self-configuration management strategy for the battlefield networks. The paper also presents a proof-of-concept simulation model that has been developed to demonstrate the viability and salient features of the proposed design in Section 5, and discusses and provides a method to estimate the performance of the proposed design in Section 6.

Regarding future work, the next step is analysis of the proposed self-configuration management system via the performance measures described in Section 6. Another item which merits future work is continuous RAP movements i.e., extend the discrete plug-and-play movements to support continuous on-the-move operations of the RAP units. In this context, we remark that the proposed RAP-
Core switch discover protocol can be shown to accommodate continuous movements. Finally, larger and hierarchical networks that utilize the scalability features of PNNI also merit future work.

8. REFERENCES


